

Weather Forecasting Software Software Requirements Specification

Version: 1.1



Signatures

| Date | Revision | Approved By |
|------------|----------|-----------------------------------|
| 11 02 2019 | 1.1 | Gagan Deep Singh and Onkar Sharma |

List of Contributors

| Name | Initials | Email |
|------------------|----------|--|
| Gagan Deep Singh | GS | gagandeep.singh2017a@vitstudent.ac.in |
| Onkar Sharma | OS | onkarsharma2017@vitstudent.ac.in |

Change History

| Revision | Date | Description |
|----------|------------|----------------------|
| 1.0 | 10 02 2019 | Initial Revision |
| 1.1 | 11 02 2019 | First Final Revision |

Preface

This document represents the Software Requirements Specification for the Weather forecasting software project. The document begins with an Introduction section that describes the purpose of the document and what is considered to be in the scope of this document and what is outside the scope of this document.

The next section is an Overall Description of the requirements and functions. This section includes the overall constraints that the project is working within as well as the assumptions made by the project as far as the defining the requirements is concerned. Lastly, the project dependencies are also listed in this section.

The Specific Requirements section is the most important section of this document. This section goes into detail about each specific requirement of the Electronic Stamp project. The Specific Requirements section also describes the Performance Requirements that are to be met by the Weather forecasting software. Design Constraints and Standards Compliance are also considered in this section.

Table of Contents

| | |
|--|----|
| Weather Forecasting Software | 1 |
| Software Requirements Specification | 1 |
| Version: 1.1..... | 1 |
| 1 Introduction..... | 1 |
| 1.1 Purpose | 1 |
| 1.2 Scope | 1 |
| 1.3 Definitions, Acronyms, and Abbreviations | 2 |
| 1.4 References | 2 |
| 2 Overall Description | 3 |
| 2.1 Product Perspective..... | 3 |
| 2.2 Product Functions..... | 4 |
| 2.3 Constraints..... | 6 |
| 2.4 User Characteristics | 6 |
| 2.5 Entity Relationships | 7 |
| 2.6 Data Flows..... | 7 |
| 2.7 Data Dictionary | 8 |
| 2.8 Assumptions | 10 |
| 3 Specific Requirements..... | 11 |
| 3.1 System Features..... | 11 |
| 3.1.1 Input current parameters | 11 |
| 3.1.2 Get predictions | 12 |
| 3.1.3 Input bulk data..... | 13 |
| 3.1.4 Add/Delete or modify data..... | 15 |
| 3.2 Performance Requirements | 16 |
| 3.3 Software System Attributes..... | 16 |
| 3.3.1 Reliability..... | 16 |
| 3.3.2 Security | 17 |
| 3.3.3 Easy to use | 17 |
| 3.3.4 Speed..... | 17 |

List of Figures

| | |
|---|---|
| Figure 1 Entity Relationship Diagram..... | 7 |
| Figure 2 Data Flow Diagram | 8 |

List of Tables

| | |
|--|----|
| Table of Definitions, Acronyms, and Abbreviations..... | 2 |
| Table of References..... | 2 |
| Table of Shall Requirements..... | 4 |
| Table of Design Constraints..... | 6 |
| Table of User Characteristics..... | 6 |
| Table of Data Dictionary | 9 |
| Table of Performance Requirements | 16 |

1 Introduction

1.1 Purpose

The purpose of the Software Requirements Specification is to describe the specific requirements of the project that are to be met. Included with the description of the requirements is a description of any constraints or assumptions that the project is working within.

This document also provides a description of any project dependencies that need to be explicitly expressed. Along with the requirements descriptions, it is also the purpose of this document to describe any performance requirements that need to be met. Any standards that are needed to be considered when developing the software are listed.

This document communicates the system attributes of the weather forecasting software. These system attributes include reliability, availability, scalability and maintainability.

1.2 Scope

It is within the scope of the project that the weather data is analyzed and stored in database. This project will display predictions on weather at a particular location, selected by user. The data will be verified for its correctness. System administrator will be provided provision to add/ extend data to data based and also to modify or remove certain data. Applications of this project are in fields of agriculture, forestry and general predictions for citizens. As the predictions made by this software are based only on limited parameters and omitted infeasible computation for regular computer systems, the software can be proved unreliable for military, navy, Air Traffic Control and marine applications. But will have the provision to advance to such applications. Hence these fields are out of the scope of this weather forecasting software.

1.3 Definitions, Acronyms, and Abbreviations

Table of Definitions, Acronyms, and Abbreviations

| Definition, Acronym, or Abbreviation | Description |
|---|---|
| SRS | Software Requirements Specification. |
| UID | Unique Identification |
| IEEE | Institute of Electrical and Electronics Engineers |

1.4 References

Table of References

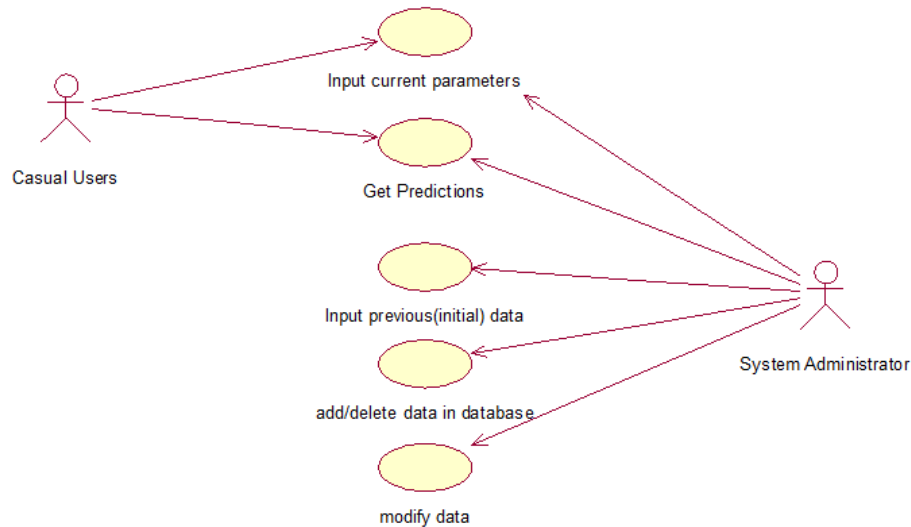
| References | Description |
|---------------------------|--|
| Software Development Plan | IEEE Std 1233, 1998 Edition, IEEE Guide for Developing System Requirements Specifications. |

2 Overall Description

2.1 Product Perspective

The project is mainly about large amounts of data that is generated by various systems in today's world and convert it to useful information for a variety of users. The Software collects weather data, such as temperature, humidity, wind speed, and various other parameters, from different sources and display as well as predict the environmental conditions of a particular location in the near future depending on the user (causal users who may use the software for general information). Recent weather patterns are analyzed to make a prediction. A user interface is provided for the users, along with the administrator. Authorization is done using a user ID and password. The users are also able to enter the current data if it is available with them. This is just to get a better idea of the real time conditions of that particular place. This data will be verified by tallying it against other sources and then added to the database. Also, the administrator can insert, modify or remove data manually from the database if there are inconsistencies along with the ability to block certain users.

The use case diagram of the software is shown below.



2.2 Product Functions

The follow is a table of the requirements that the project SHALL meet.

Table of Shall Requirements

| ID | Stakeholders | Shall Requirement |
|----|--------------|---|
| 01 | User | <p>The user SHALL be able to sign-up himself/herself using his/her general information.</p> <p>The user SHALL be provided with login ID and a password</p> <p>The user SHALL be able to login using his or her ID and password.</p> <p>The user SHALL be able to reset his or her password incase he or she forgets it.</p> <p>The user SHALL be able to select location of his or her choice about which he or she wants to see the weather information.</p> <p>The user SHALL be able to input the data that is available to him or her using trusted instruments.</p> <p>The user SHALL be able to get information about weather at his or her selected and available locations based on data available.</p> <p>The user SHALL be able to see the percentage of correctness or likely hood of occurring the events predicted by the software.</p> <p>The user SHALL be able to interact with the administrator, formally, about the data and predictions.</p> <p>The user SHALL be able to give feedback on each prediction requested.</p> |

| ID | Stakeholders | Shall Requirement |
|----|---------------|--|
| 02 | Administrator | <p>The Administrator SHALL be able to act as a casual user.</p> <p>The Administrator SHALL be able to sign-up with his/her details.</p> <p>The Administrator SHALL be provided with a unique ID and a password.</p> <p>The Administrator SHALL be able to sign in using his/her credentials.</p> <p>The Administrator SHALL be able to enter previous weather data of a particular location, thereby adding that corresponding location in the software.</p> <p>The Administrator SHALL be able to verify the data entered by the users.</p> <p>The Administrator SHALL be add/delete particular data if found inconsistent, manually.</p> <p>The Administrator SHALL be able to modify data based on predefined criteria if necessary.</p> <p>The Administrator SHALL be able to answer/comment on the questions of the users.</p> <p>The Administrator SHALL be able to remove a particular location, if the data collected from that location found inconsistent or proved to be wrong.</p> |

2.3 Constraints

The follow is a table of the design constraints that the system SHALL meet.

Table of Design Constraints

| ID | Stakeholders | Shall Requirement |
|----|------------------------|---|
| 01 | System (User) | <p>The system SHALL be able to prevent user to choose invalid or used IDs and passwords.</p> <p>The system SHALL be able to prevent user to perform any type of SQL injection attacks.</p> <p>The system SHALL be able to prevent user to login into the software without valid authentication.</p> <p>The system SHALL be able to verify the person in case the person forgets his or her password.</p> <p>The system SHALL be able to prevent user to add data from untrusted source.</p> <p>The system SHALL be able to prevent user to modify or delete data once verified.</p> |
| 02 | System (administrator) | <p>The system SHALL be to prevent any user to login as administrator.</p> <p>The system SHALL be to prevent any kind of attacks on administrator's data.</p> <p>The system SHALL be able to prevent invalid authentication and grant access to wrong user.</p> <p>The system SHALL be to prevent administrator to add or delete data without verification, checking correctness of data and its format.</p> <p>The system SHALL be to prevent administrator to modify data if there is no inconsistency found with it.</p> <p>The system SHALL be able to prevent administrator to delete a location without a valid reason based on consistency and correctness of data.</p> |

2.4 User Characteristics

The following table identifies and describes the different users of the weather forecasting software. The information gathered about the different users of the system helped define what the software needs to do. Also, these users are referenced in the requirements and diagrams.

Table of User Characteristics

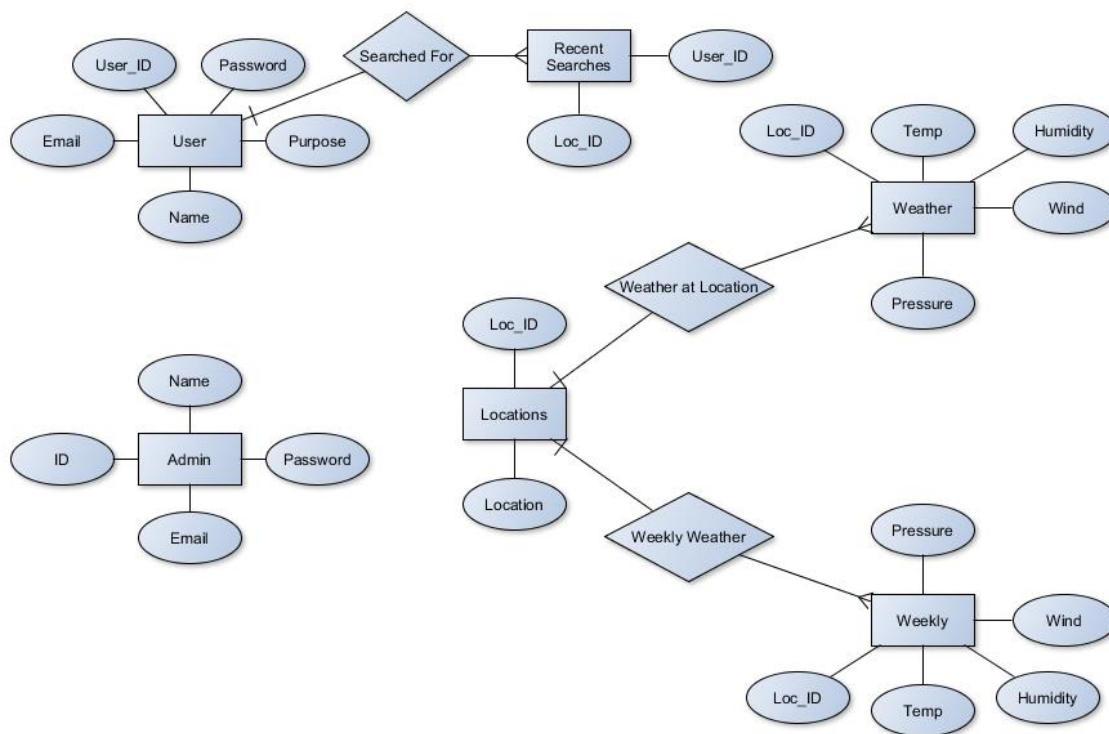
| User | Description |
|--------------|---|
| Causal users | The user is any one who is using the system for general prediction of weather data at a particular location. He has a |

| | |
|---------------|--|
| | small role in adding small piece of data to the database after verification by the concerned administrator. |
| Administrator | The Administrator user will be computer literate and technically competent in performing administration on computer systems. |

2.5 Entity Relationships

Figure 1 shows the entity relationships for the Weather forecasting software project.

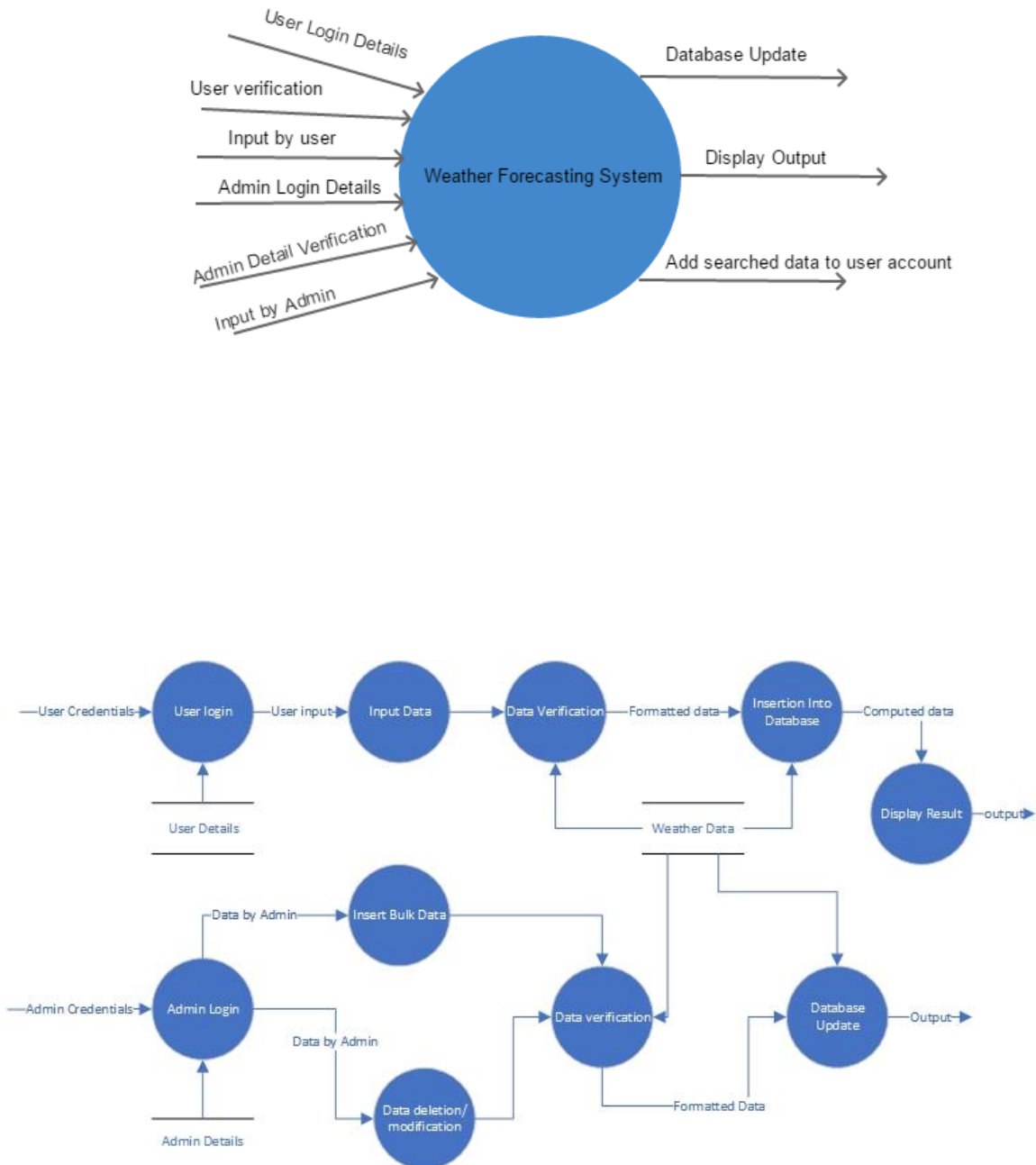
Figure 1 Entity Relationship Diagram



2.6 Data Flows

The following figures represent the data flow diagrams of the weather forecasting software. The first data flow diagram, figure 2, is the top-level data flow. This is followed by the more detail data flow of the Software.

Figure 2 Data Flow Diagram



2.7 Data Dictionary

The follow tables in this section make up the data dictionary for the weather forecasting software project. Using the Data Flow Diagrams, the following Data Dictionary elements were defined.

- User Details
- Admin Details
- Raw Data
- Segregated data
- Recent Searches

Table of Data Dictionary

| Data Dictionary Attribute | Detail |
|---------------------------|---|
| Name | User Details |
| Aliases | User Account Details/User Credentials |
| Where Used / How Used | Create User Account (Input) User Login (Input/Output) Delete Account (Output) Recent Searches (Automatic Input) |
| Description | User details = User_ID + Password + Name + Email + Purpose of Use User_ID = Unique id of characters (length=8) Password = String containing upper case, lower case, numbers and special characters and the length is between 8 – 16 characters. |
| Supplementary Information | It is majorly used for user authentication and keeping a record of recent searches. Can also be used to provide user with user specific data. |

| Data Dictionary Attribute | Detail |
|---------------------------|---|
| Name | Admin Details |
| Aliases | Admin Account Details/Admin Credentials |
| Where Used / How Used | Admin Login (Input/Output) Delete Account (Output) Data Modification (Input) |
| Description | Admin details = ID + Password + Name + Email ID = Unique id of characters (length=8) Password = String containing upper case, lower case, numbers and special characters and the length is between 8 – 16 characters. |
| Supplementary Information | Is used for by a site admin for login. Bulk Insertion of data as well as modification or addition to the database requires admin authentication. |

| Data Dictionary Attribute | Detail |
|---------------------------|--|
| Name | Raw Data |
| Aliases | Input by User |
| Where Used / How Used | User input (Input) Data verification (Input/Output) Data Formatting (Input/Output) Database Entry (Input) |
| Description | Raw Data = Location + Weather Info (Temperature, Humidity, Wind speed and direction, Pressure) |
| Supplementary Information | A raw input by the user in order to get predictions. |

| | |
|--|--|
| | |
|--|--|

| Data Dictionary Attribute | Detail |
|---------------------------|---|
| Name | Segregated Data |
| Aliases | Verified data/Formatted data |
| Where Used / How Used | Insertion into Database (Input) Weekly Average Calculations (Input) Display to the User (Output) |
| Description | Segregated Data = Raw Data + Arrangement according to week + Proper Formatting |
| Supplementary Information | The raw data is formatted properly, is segregated according to week after the average of each of the attributes is calculated for every week. |

| Data Dictionary Attribute | Detail |
|---------------------------|--|
| Name | Recent Searches |
| Aliases | None |
| Where Used / How Used | Display User his/her past few searches (Output) User specific data (Input) |
| Description | The past few searches, say 10, is stored according to the location IDs. This is displayed directly to the user or can be used by the system to calculate user specific data. |
| Supplementary Information | None |

2.8 Assumptions

The following table lists the assumptions made by the requirements that define the weather forecasting software.

| Assumption | Description |
|-------------------|---|
| Hardware quality. | One assumption about the product is that it will always be implemented and used on hardware devices that is the weather measuring instruments that have good performance. |

3 Specific Requirements

3.1 System Features

3.1.1 Input current parameters

3.1.1.1 Introduction

The Weather forecasting software shall allow a user to input the data available to him/her. The user may gather the data from any trusted sources or instruments. The data will be verified and get added to database.

3.1.1.2 Functional Requirements

Purpose: Keep adding data to database for better predictions in future.

Input: Weather data like temperature, humidity, wind, etc. The data should be in supported format and encoding.

Processing: Checking the encoding and formatting of the data. Translating data in common format. Checking validity of the data.

Output: The data entered and verified gets added to the database and administrator is informed about the same.

3.1.1.3 Stimulus Response

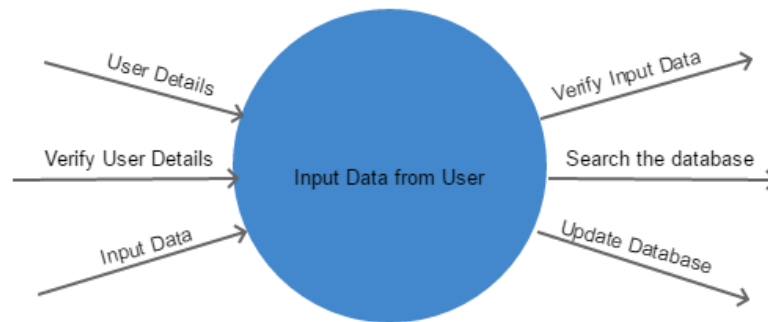
A) User enters the data in user friendly format.

| User Actions | System Actions |
|-------------------------------------|---------------------------------------|
| (1) Input the data | |
| (2) Submit the source or instrument | |
| | (3) Check the formatting and encoding |
| | (4) Check validity of the data |
| (5) Receive validity status | |
| | (6) Enter the data into database |

B) User enters the data in format or encoding that is alien to the system.

| User Actions | System Actions |
|-------------------------------------|--|
| (1) Input the data | |
| (2) Submit the source or instrument | |
| | (3) Check the formatting and encoding |
| | (4) Change the formatting and encoding |

| | |
|-----------------------------|--------------------------------|
| | into desired format. |
| | (5) Check validity of the data |
| (6) Receive validity status | |
| | (7) Enter data into database |



3.1.2 Get predictions

3.1.2.1 Introduction

The software performs calculations on the database and predict the weather conditions of near future.

3.1.2.2 Functional Requirements

Purpose: Calculate and display weather information at desired and available locations.

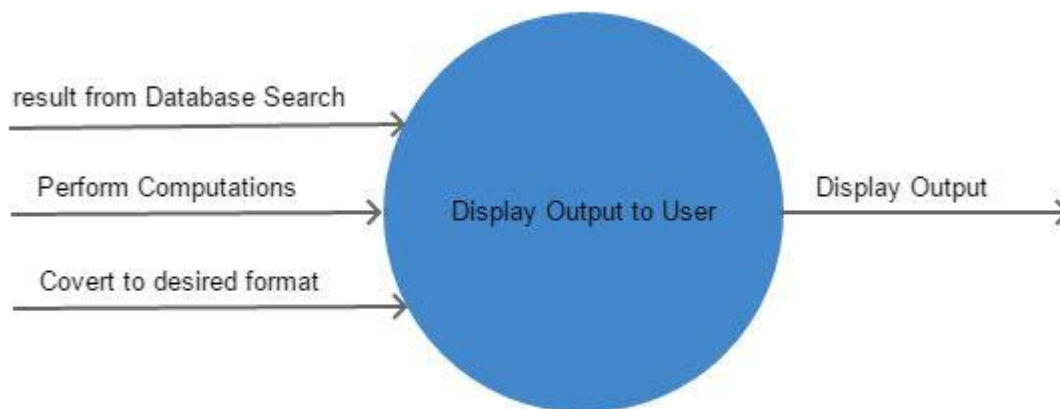
Input: Command to display weather predictions.

Processing: Mathematical and statistical calculations on the data and display the result.

Output: Weather predictions of not too distinct future at desired and available location gets displayed.

3.1.2.3 Stimulus Response

| User Actions | System Actions |
|-----------------------------------|---|
| | (1) Calculations are performed on the data available in the database. |
| | (2) Weather predictions are made |
| | (3) Validity and correctness are checked |
| | (4) New predictions are included in the database |
| | (5) Predictions are displayed along with correctness. |
| (6) Mark data to be useful or not | |



3.1.3 Input bulk data

3.1.3.1 Introduction

The Weather forecasting software shall allow the administrators to input the data available to him/her. They may gather the data from any trusted sources or instruments. The data is in bulk will be verified and get added to database.

3.1.3.2 Functional Requirements

Purpose: Initializing the database as weather predictions require large amount of data.

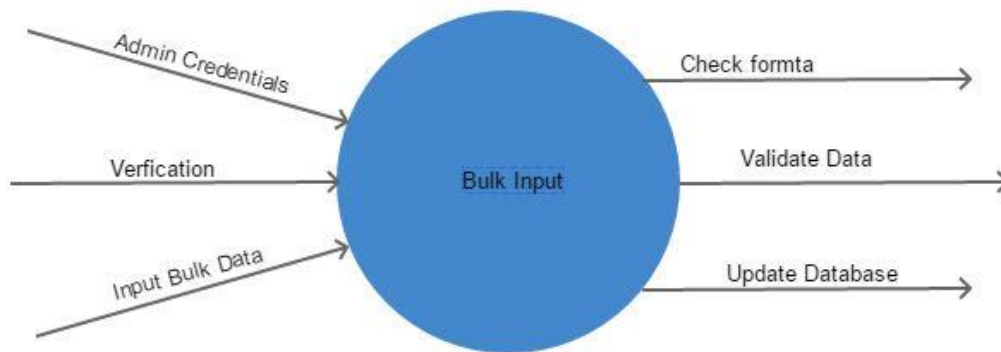
Input: Weather data like temperature, humidity, wind, etc. The data should be in supported format and encoding.

Processing: Checking the encoding and formatting of the data. Translating data in common format. Checking validity of the data.

Output: The data entered and verified gets added to the database and activity log gets updated.

3.1.3.3 Stimulus Response

| Administrator Actions | System Actions |
|-------------------------------------|---|
| (1) Input the data | |
| (2) Submit the source or instrument | |
| | (3) Check the formatting and encoding |
| | (4) Change the formatting and encoding into desired format. |
| | (5) Check validity of the data |
| (6) Receive validity status | |
| | (7) Enter data into database |



3.1.4 Add/Delete or modify data

3.1.4.1 Introduction

The data in the database can be proved inconsistency or wrong. The administrator can delete such data that is always recommended. In some cases, if the data is proved to be wrong and is essential for calculations then data can be modified manually.

3.1.4.2 Functional Requirements

Purpose: Deleting or modifying the data.

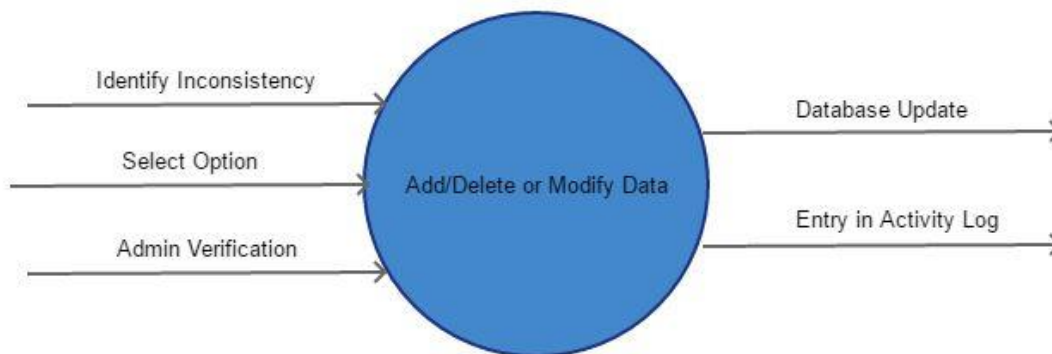
Input: Administrator mark the inconsistent data and select require options whether to delete or to change the data.

Processing: Database get updated.

Output: Acknowledgment of the action is generated and activity log gets updated and is displayed.

3.1.4.3 Stimulus Response

| Administrator Actions | System Actions |
|---|--|
| (1) Inconsistent data is marked | |
| (2) Desired option is selected. | |
| (3) Action and administrator verify himself/herself | |
| | (4) Data in the database gets modified |
| | (5) Activity log is updated. |



3.2 Performance Requirements

The following tables list the performance requirements of the Weather forecasting software.

Table of Performance Requirements

| Performance Requirement | Description |
|-------------------------|--|
| Data Storage Capacity | The data storage capacity of the Weather forecasting software will be configurable to the extent that other generic data analyzers software allows and the hardware permits. |
| Software Runtime Errors | The Weather forecasting software will handle the runtime errors consistently and as gracefully as possible. |

3.3 Software System Attributes

3.3.1 Reliability

Reliability in the Weather forecasting software will be ensured by thorough unit, milestone, and release testing. Comprehensive test scenarios and acceptance criteria will be established to reflect the necessary level reliability required of the Weather forecasting software. The all delivered source code will be thoroughly tested using the established test scenarios until the acceptance criteria are satisfied by the Weather forecasting software.

3.3.2 Security

To store sensitive information in database the software will use hashing techniques and various encryption algorithms. Verification and authentication of user and administrator will be done thorough secured channel and anonymity will be maintained.

3.3.3 Easy to use

Training time to use the Weather forecasting software is in 2 to 3 days that include learning al details of the software use. This time period is reasonable. However, this does not include learning internal mathematical and statistical calculations.

3.3.4 Speed

The speed or number of processed transactions per second of the weather forecasting software is also reasonable, but also machine dependent. Screen refresh time is also monitored and efforts are in reducing this time thereby increasing the speed.

